CS 173 Lecture 21c: To Infinity & Beyond 

Examples of game/puzzle diagrams were finite in 2 ways:

(1) # states was finite

2x2 tic-tac-toe: 29 states

jugs puzzle: 24 states

(2) each state had bounded complexity

2x2 tic-tactoe: each state was four values (one for each cell) w/ possibilities X, O, blank

jugs puzzle: each state was two numbers between 0 & 5.

Euclid GCD: infinite # of states: one for each pair $(x,y) \in \mathbb{N} \times \mathbb{N}$.

for each # of digits k, there is a pair (x,y) needing $\geq k$ digits to write.

In general, computer programs need infinity in both directions.

Conway's Game of Life

↳ "Turing-complete"

↳ every "standard" program can be simulated w/ GoL.

can be thought of via state diagrams necessarily infinite

Rules: "cells" on $\mathbb{Z} \times \mathbb{Z}$

birth: dead cell adjacent to 3 live cells comes

Rules: "cells" on $\mathbb{Z} \times \mathbb{Z}$

transitions

- birth: dead cell adjacent to 3 live cells comes to life (left, right, up, down, diagonal)
- life: a live cell adjacent to 2 or 3 live cells stays alive
- death: a live cell adjacent to <2 or >3 live cells dies

state: list of live cells.

- infinite # of states $2^{|\mathbb{Z} \times \mathbb{Z}|}$???
- states have unbounded complexity
  for every k, have $\geq k$ live cells.